

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-330419

(43)Date of publication of application : 22.12.1997

-----  
(51)Int.Cl. G06T 11/00

G06F 17/50

G09G 5/36

G09G 5/36

-----  
(21)Application number : 08-168421 (71)Applicant : NIPPON STEEL CORP

(22)Date of filing : 07.06.1996 (72)Inventor : MATSUURA KAZUNORI

-----  
(54) GRAPHIC AREA DISCRIMINATING DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To accelerate a graphic area discriminating processing based on an algorithm by discriminating an area including a graphic based on output information of an AND gate and an OR gate.

SOLUTION: Logical codes respectively assigned to areas constituting a graphic are obtained as the vertex codes of respective vertexes by an ALU 1 to store in a shift register 4. The AND and OR of the respective vertex codes stored in the shift register 4 are obtained by an AND gate means 5 and an OR gate means 6. Then based on output information of the means 5 and 6,, the area including the graphic is discriminated by a condition judging part 8. Thereby the graphic area discriminating processing based on an algorithm is accelerated.

-----  
LEGAL STATUS [Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

\* NOTICES \*

JPO and INPIT are not responsible for any

damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.

2.\*\*\*\* shows the word which can not be translated.

3.In the drawings, any words are not translated.

---

## CLAIMS

---

[Claim(s)]

[Claim 1] In the graphic form field distinction equipment for distinguishing where [ in two or more fields which change in a viewing area and the non-display field in the perimeter ] the graphic form exists A top-most-vertices code calculation means to ask for the logical code currently assigned to the field to which each top-most vertices which constitute the above-mentioned graphic form belong, respectively as a top-most-vertices code of each top-most vertices, and to store in a register the top-most-vertices code of each top-most vertices for which it asked, The AND gate which takes the AND of the top-most-vertices code of each top-most vertices stored in the above-mentioned register by the above-mentioned top-most-vertices code calculation means, The OR gate which takes the OR of the top-most-vertices code of each top-most vertices stored in the above-mentioned register by the above-mentioned top-most-vertices code calculation means, Graphic form field distinction equipment characterized by

having a field distinction means to distinguish in which field the above-mentioned graphic form exists, based on the print-out of the above-mentioned AND gate and the above-mentioned OR gate.

[Claim 2] The comparison means which the above-mentioned top-most-vertices code calculation means carries out the sequential comparison of the coordinate value of each top-most vertices which constitute the above-mentioned graphic form, and the boundary value of the viewing area which exists at the core of two or more above-mentioned fields for every top-most vertices, and carries out the sequential output of the flag information according to the comparison result, Graphic form field distinction equipment according to claim 1 characterized by consisting of shift registers which obtain the top-most-vertices code of each above-mentioned top-most vertices by storing carrying out the sequential shift of the flag information by which a sequential output is carried out from the above-mentioned comparison means.

[Claim 3] While establishing further the 1st OR gate which takes the OR of each bit which constitutes the AND of each top-most-vertices code called for by the above-mentioned AND gate in the latter part of the above-mentioned AND gate The 2nd OR gate which takes the OR of each bit which constitutes the OR of

each top-most-vertices code for which the above-mentioned OR gate asked is further established in the latter part of the above-mentioned OR gate. The above-mentioned field distinction means It will distinguish, if the above-mentioned graphic form is in a perfect invisible field when the print-out of the 1st OR gate of the above is "1." if the above-mentioned graphic form is in a perfect visible field when the print-out of the 2nd OR gate of the above is "0" -- distinguishing -- time it is other -- the above-mentioned graphic form -- perfect invisibility or a part -- the graphic form field distinction equipment according to claim 1 or 2 characterized by distinguishing if it is in a visible field.

[Claim 4] In the graphic form field distinction equipment for distinguishing where [ in two or more fields which change in a viewing area and the non-display field in the perimeter ] the graphic form exists The comparison means which carries out the sequential comparison of the coordinate value of each top-most vertices which constitute the above-mentioned graphic form, and the boundary value of the viewing area which exists at the core of two or more above-mentioned fields for every top-most vertices, and carries out the sequential output of the flag information according to the comparison result, The shift register which obtains the top-most-vertices code of each top-most vertices by storing carrying out the

sequential shift of the flag information by which a sequential output is carried out from the above-mentioned comparison means, The AND gate means constituted so that the AND code of "1" or "0" might be outputted by whether there is at least one bit of "1" in each bit which constitutes the AND of each top-most-vertices code stored in the above-mentioned shift register, The OR-gate means constituted so that the OR code of "1" or "0" might be outputted by whether there is at least one bit of "1" in each bit which constitutes the OR of each top-most-vertices code stored in the above-mentioned shift register, It will distinguish, if the above-mentioned graphic form is in a perfect invisible field when the AND code outputted from the above-mentioned AND gate means is "1." if the above-mentioned graphic form is in a perfect visible field when the OR code outputted from the above-mentioned OR-gate means is "0" -- distinguishing -- time it is other -- the above-mentioned graphic form -- perfect invisibility or a part -- the graphic form field distinction equipment characterized by having a field distinction means to distinguish if it is in a visible field.

[Claim 5] The space where the above-mentioned graphic form exists is graphic form field distinction equipment given in any 1 term of claims 1-4 characterized by for the above-mentioned viewing area being a rectangle and being the

two-dimensional space where the above-mentioned non-display field was divided into eight pieces to the direction of four directions of the above-mentioned viewing area.

[Claim 6] The space where the above-mentioned graphic form exists is graphic form field distinction equipment given in any 1 term of claims 1-4 characterized by for the above-mentioned viewing area being a rectangular parallelepiped, and being the three-dimension space where the above-mentioned non-display field was divided into 28 pieces to the four-directions cross direction of the above-mentioned viewing area.

---

## DETAILED DESCRIPTION

---

[Detailed Description of the Invention]

[0001]

[Field of the Invention] In case this invention pretreats clipping performed in fields, such as computer graphics, concerning graphic form field distinction equipment, it is used, and it is suitable.

[0002]

[Description of the Prior Art] In fields, such as computer graphics (CG), in case the body made from CG is displayed, clipping processing is performed in many cases. In case a body is displayed on a display, clipping sets up the fixed viewing area, and in order to make it not display the part protruded out of the viewing area, it means the thing of processing which excepts the part in the exterior of a viewing area.



[0003] Although the polygon which constitutes the two-dimensional body and three-dimension body of CG is constituted by the graphic form of a triangle, a square, etc. What is in \*\* completeness inside a viewing area (clipping window) 50 as shown in the graphic form at drawing 5 (the perfect visible triangle A), \*\* There are three kinds of what is in the outside of a viewing area 50 thoroughly (perfect invisible triangles B and D), and the thing (part the visible triangle C) divided into the inside and an outside by the boundary of the \*\* viewing area 50. the invisible triangles B and D perfect in the example of this drawing 5 at clipping processing, and a part -- processing which excepts a part of visible triangle C is performed.

[0004] Thus, it is because the load of subsequent processing will become large if performing clipping processing leaves the graphic form which is not displayed as it is.

[0005] In order to perform the above clipping processings, it is necessary to distinguish whether a graphic form exists in the field of above-mentioned \*\* - \*\* throat as the pretreatment. There are some which are called a Cohen Sutherland algorithm to one of the algorithms which distinguish three kinds of this field. Below, the Cohen Sutherland algorithm applied to two-dimensional CG is

explained using drawing 5 .

[0006] First, a 4-bit logical code is assigned to each of nine fields divided by the clipping straight line (four borderlines) which constitutes the clipping window 50.

Although the logical code in the clipping window 50 is "0000", "1" is assigned to the least significant bit in the field on the left of it, and "1" is assigned to the 2nd bit in a right-hand side field. Moreover, "1" is assigned to the 3rd bit in the field below the clipping window 50, and "1" is assigned to the most significant bit in an upper field.

[0007] Next, it investigates in any in nine fields which each top-most vertices of a polygon (the example of drawing 5 triangle) described above it exists, and a top-most-vertices code is assigned to each top-most vertices. In the case of the example of drawing 5 , the top-most-vertices code of triangle A-D is as follows, respectively.

"0000" triangle A: "0000", "0000", and triangle B: "1001", "1001", "1000" triangle

C: "0010", "0000", and "0010" triangle D: "1010", "1000", "0010" [0008] And it

classifies into three groups by performing the condition judging shown below using each top-most-vertices code of the triangle assigned as mentioned above.

(1) the triangle is perfect if the OR of three top-most-vertices codes is "0000" --

visible (triangle A).

(2) The triangle is perfect invisible if the AND of three top-most-vertices codes is not "0000" (triangle B).

(3) things other than the above (1) and (2) -- perfect invisibility or a part -- visible (a triangle C and a triangle D).

[0009] in addition, this Cohen Sutherland algorithm showed to the above (2) and (3) -- as -- a perfect invisible graphic form and a part -- although a visible graphic form cannot be classified thoroughly, any graphic form is set as the object of clipping processing, and an exact classification is performed in subsequent clipping processing. Although the above was clipping pretreatment in two-dimensional space, since the element of order other than an vertical and horizontal element is added in three-dimension space, a logical code and a top-most-vertices code become 6 bits.

[0010] Drawing 3 is drawing showing the configuration of the conventional graphic form field distinction equipment which performs graphic form field distinction based on the above-mentioned Cohen Sutherland algorithm, and drawing 4 is drawing showing the example of the conventional program for performing this graphic form field distinction. Hereafter, it explains using these

drawings.

[0011] In addition, the program which performs the program of drawing 4 by the usual processors (DSP etc.) was shown, and it is assumed that the clipping window 50 is a cube which becomes in the range of  $-W \leq X \leq W$ ,  $-W \leq Y \leq W$ , and  $-W \leq Z \leq W$ . However, X, Y, and Z are the coordinate values of the longitudinal direction which made the core of the clipping window 50 the starting point, the vertical direction, and a cross direction, respectively. Moreover, the coordinate of each top-most vertices of the polygon under processing is set to (X [i], Y [i], Z [i]) (1 i= 0, 2).

[0012] It is the top-most-vertices code vertex of three top-most vertices where the part of (a) constitutes a triangle in drawing 4 [i]. (1 i= 0, 2) It is 0(10)=000000(2) altogether. It is the processing to initialize. Moreover, the part of (b) is each top-most-vertices code vertex [i]. It is the processing for which it asks actually. When performing graphic form field distinction in three-dimension space, it is necessary to perform six condition judgments to ask for one top-most-vertices code.

[0013] The 1st condition judging is the X coordinate value X of a certain top-most vertices [i]. It is the processing which judges whether it is smaller than -W which

is a boundary value on the left-hand side of the clipping window 50. the case where this condition is satisfied -- top-most-vertices code vertex at that time [i] (what was initialized) 1(10) = -- 000001 (2) Top-most-vertices code vertex with what [ new ] took the OR [i] \*\* -- by being carried out, "1" is stood to the least significant bit.

[0014] The 2nd condition judging is the above-mentioned X coordinate value X [i]. When it is the processing which judges whether it is larger than W which is a boundary value on the right-hand side of the clipping window 50 and this condition is satisfied top-most-vertices code vertex at that time [i] 2(10) = -- 000010 (2) Top-most-vertices code vertex with what [ new ] took the OR [i] \*\* -- by being carried out, "1" is stood to the 2nd bit.

[0015] the following -- the same -- carrying out -- Y coordinate value Y [i] Z coordinate value Z [i] about -- a condition judging is performed one by one. As Notation or shows in the condition judging over these 6 times, it is the last top-most-vertices code vertex [i]. Since the OR is taken with the value, "1" is stood to all the bits corresponding to the concluded conditional expression. By this six condition judging processing, it is the top-most-vertices code vertex about one top-most vertices [i]. It asks. They are three top-most-vertices codes

vertex [i] by repeating such processing 3 times and performing it, when the polygon which constitutes a three-dimension body is a triangle. (1 i= 0, 2) It asks.

[0016] Thus, top-most-vertices code vertex [i] ALU (arithmetic and logic unit)31 of drawing 3 is performing processing of drawing 4 (a) and (b) for which it asks.

At this time, the boundary value W and -W which are used by six condition judging processings are stored in one register 32, and it is each coordinate value X [i]. Y [i] Z [i] It is stored in the register 33 of another side.

[0017] Moreover, it is each top-most-vertices code vertex [i] of the triangle which asked for the parts of drawing 4 (c) and (d) as mentioned above. (1 i= 0, 2) It used and described above. (1) - (3) It is the processing which performs a group division. That is, in the part of drawing 4 (c), they are three top-most-vertices codes vertex [0]. vertex [1] vertex [2] An AND and an OR are searched for and three kinds of group divisions are performed in the part of drawing 4 (d) by seeing whether the AND searched for is 0 (10) and the OR searched for whether be 0 (10).

[0018] ALU31 of drawing 3 performs processing of above-mentioned drawing 4 (c), and ALU31 and the condition judging section 35 of drawing 3 are performing processing of drawing 4 (d). That is, they are three top-most-vertices codes

vertex [i] which calculated ALU31 by processing of drawing 4 (b) first. (1 i= 0, 2)

It inputs through registers 32 and 33 and those ANDs and ORs are searched for.

Next, the polygon (triangle) which the value of the AND searched for is not 0 (10),

or corresponds is the above (2). It investigates whether it is a thing belonging to

a group.

[0019] In addition, on the property of count of drawing 4 (b), since an AND does

not take a negative value, it is the same as an AND being larger than 0 (10) that

an AND is not 0 (10). Therefore, it is the above (2) by seeing whether as shown

in the maximum upper case of drawing 4 (d), the AND searched for is larger than

0 (10). It has judged as follows whether it belongs to a group.

[0020] First, ALU31 adds 0 (10) to the value of the AND searched for the

account of a top. At this time, if the value acquired by addition processing is 0

(10), "1" of a zero flag will be stood to the sign flag register 34. Therefore, if the

value of the sign flag register 34 is "0", it will be said that the AND searched for is

larger than 0 (10). Then, the condition judging section 35 investigates whether

the AND searched for is larger than 0 (10) by seeing the value of the

above-mentioned sign flag register 34.

[0021] Since I hear that the value of the AND searched for is not 0 (10) when it

judges that the value of the above-mentioned sign flag register 34 is "0", the condition judging section 35 is controlled to choose the address value to which only 1 incremented the value of a program counter 37 by the selector 36, and performs processing it is supposed that the corresponding triangle is a perfect invisible thing. On the other hand, when it judges that the value of the above-mentioned sign flag register 34 is "1", it controls to choose the address value of a branching place by the selector 36 in order to perform the condition judging about an OR next.

[0022] Thus, when a program is advanced to the branching place address which performs the condition judging of an OR next, for ALU31 and the condition judging section 35, whether the value of the OR searched for being 0 (10) as shown in the 2nd step of drawing 4 (d), and the corresponding polygon (triangle) are the above (1). It investigates whether it is a thing belonging to a group. When the condition judging section 35 looks at the value of the sign flag register 34 obtained when 0 (10) is added to the value of an OR also in this case in ALU31, it investigates whether the value of the OR searched for is 0 (10).

[0023] Since I hear that the value of the OR searched for is 0 (10) when it judges that the value of the above-mentioned sign flag register 34 is "1", the condition



judging section 35 is controlled to choose the address value to which only 1 incremented the value of a program counter 37 by the selector 36, and performs processing which the corresponding triangle makes completely visible.

[0024] the triangle which controls to choose the address value of a branching place by the selector 36, advances a program to the branching place in order to process the bottom of drawing 4 (d), and corresponds on the other hand when it judges that the value of the above-mentioned sign flag register 34 is "0" -- perfect invisibility or a part -- processing made visible is performed.

[0025]

[Problem(s) to be Solved by the Invention] As shown in drawing 4., in order to distinguish whether the polygon exists in which field based on a Cohen Sutherland algorithm, much instructions are executed with conventional graphic form field distinction equipment. Therefore, activation of the above-mentioned algorithm had taken long time amount. Then, this invention aims at enabling it to perform more graphic form field distinction processing based on the above-mentioned Cohen Sutherland algorithm at a high speed.

[0026]

[Means for Solving the Problem] In graphic form field distinction equipment for

the graphic form field distinction equipment of this invention to distinguish where [ in two or more fields which change in a viewing area and the non-display field in the perimeter ] the graphic form exists A top-most-vertices code calculation means to ask for the logical code currently assigned to the field to which each top-most vertices which constitute the above-mentioned graphic form belong, respectively as a top-most-vertices code of each top-most vertices, and to store in a register the top-most-vertices code of each top-most vertices for which it asked, The AND gate which takes the AND of the top-most-vertices code of each top-most vertices stored in the above-mentioned register by the above-mentioned top-most-vertices code calculation means, The OR gate which takes the OR of the top-most-vertices code of each top-most vertices stored in the above-mentioned register by the above-mentioned top-most-vertices code calculation means, It is characterized by having a field distinction means to distinguish in which field the above-mentioned graphic form exists, based on the print-out of the above-mentioned AND gate and the above-mentioned OR gate.

[0027] The place by which it is characterized [ of this invention / other ] the above-mentioned top-most-vertices code calculation means The comparison means which carries out the sequential comparison of the coordinate value of

each top-most vertices which constitute the above-mentioned graphic form, and the boundary value of the viewing area which exists at the core of two or more above-mentioned fields for every top-most vertices, and carries out the sequential output of the flag information according to the comparison result, It is characterized by consisting of shift registers which obtain the top-most-vertices code of each above-mentioned top-most vertices by storing carrying out the sequential shift of the flag information by which a sequential output is carried out from the above-mentioned comparison means.

[0028] While establishing further the 1st OR gate which takes the OR of each bit which constitutes the AND of each top-most-vertices code called for by the above-mentioned AND gate in the latter part of the above-mentioned AND gate, the place by which it is characterized [ of others of this invention ] The 2nd OR gate which takes the OR of each bit which constitutes the OR of each top-most-vertices code for which the above-mentioned OR gate asked is further established in the latter part of the above-mentioned OR gate. The above-mentioned field distinction means It will distinguish, if the above-mentioned graphic form is in a perfect invisible field when the print-out of the 1st OR gate of the above is "1." if the above-mentioned graphic form is in a

perfect visible field when the print-out of the 2nd OR gate of the above is "0" -- distinguishing -- time it is other -- the above-mentioned graphic form -- perfect invisibility or a part -- it is characterized by distinguishing, if it is in a visible field.

[0029] In graphic form field distinction equipment for the place by which it is characterized [ of others of this invention ] to distinguish where [ in two or more fields which change in a viewing area and the non-display field in the perimeter ] the graphic form exists The comparison means which carries out the sequential comparison of the coordinate value of each top-most vertices which constitute the above-mentioned graphic form, and the boundary value of the viewing area which exists at the core of two or more above-mentioned fields for every top-most vertices, and carries out the sequential output of the flag information according to the comparison result, The shift register which obtains the top-most-vertices code of each top-most vertices by storing carrying out the sequential shift of the flag information by which a sequential output is carried out from the above-mentioned comparison means, The AND gate means constituted so that the AND code of "1" or "0" might be outputted by whether there is at least one bit of "1" in each bit which constitutes the AND of each top-most-vertices code stored in the above-mentioned shift register, The

OR-gate means constituted so that the OR code of "1" or "0" might be outputted by whether there is at least one bit of "1" in each bit which constitutes the OR of each top-most-vertices code stored in the above-mentioned shift register, It will distinguish, if the above-mentioned graphic form is in a perfect invisible field when the AND code outputted from the above-mentioned AND gate means is "1." if the above-mentioned graphic form is in a perfect visible field when the OR code outputted from the above-mentioned OR-gate means is "0" -- distinguishing -- time it is other -- the above-mentioned graphic form -- perfect invisibility or a part -- it is characterized by having a field distinction means to distinguish if it is in a visible field.

[0030] The above-mentioned viewing area is a rectangle and space where the above-mentioned graphic form exists the place by which it is characterized [ of others of this invention ] is characterized by being the two-dimensional space where the above-mentioned non-display field was divided into eight pieces to the direction of four directions of the above-mentioned viewing area.

[0031] The above-mentioned viewing area is a rectangular parallelepiped, and space where the above-mentioned graphic form exists the place by which it is characterized [ of others of this invention ] is characterized by being the

three-dimension space where the above-mentioned non-display field was divided into 28 pieces to the four-directions cross direction of the above-mentioned viewing area.

[0032] Since this invention consists of the above-mentioned technical means, it can be managed, even if those ANDs and ORs come to be searched for promptly automatically and it searches for neither an AND nor an OR by count like before, when the part which searches for the AND of the top-most-vertices code of each top-most vertices, and the part which searches for an OR are constituted by an AND gate and the OR gate in hardware, respectively and the top-most-vertices code of each top-most vertices is called for.

[0033] According to other descriptions of this invention, since it asks only by carrying out the sequential transfer of the flag information on a comparison result at a shift register whenever the top-most-vertices code of each top-most vertices carries out the sequential comparison of the coordinate value of each top-most vertices, and the boundary value of a viewing area for every top-most vertices and it performs one comparison processing, compared with the former, each top-most-vertices code comes to be called for by fewer computational procedures.

[0034] Since according to the description of others of this invention the device for searching for the information referred to as conditions for a conditional-branching instruction is constituted by the 1st and 2nd OR gate in hardware in case field distinction is performed, it can be managed, even if such reference information comes to be promptly called for automatically from the AND and OR of each top-most-vertices code and it does not ask by count like before.

[0035]

[Embodiment of the Invention] Hereafter, 1 operation gestalt of this invention is explained based on a drawing. Drawing 1 is drawing showing the configuration of the graphic form field distinction equipment concerning this operation gestalt for distinguishing where [ in two or more fields which change in a viewing area and the non-display field divided into the plurality in that perimeter ] the graphic form exists, and drawing 2 is drawing showing the example of the program by this operation gestalt for performing this graphic form field distinction.

[0036] In addition, in the program of drawing 2 , the viewing area (clipping window) assumes that it is the cube which becomes in the range of  $-W \leq X \leq W$ ,  $-W \leq Y \leq W$ , and  $-W \leq Z \leq W$ . However, X, Y, and Z are the coordinate values of

the longitudinal direction which made the core of a clipping window the starting point, the vertical direction, and a cross direction, respectively. Moreover, the coordinate of each top-most vertices of the polygon under processing is set to  $(X[i], Y[i], Z[i])$  ( $i = 0, 2$ ). In this case, a non-display field is divided into the four-directions cross direction of a viewing area to 28 fields.

[0037] In drawing 1 , 1 is ALU and performs an operation which is described below using the information stored in two registers 2 and 3. That is, ALU1 performs the operation which carries out the sequential comparison of the coordinate value of each top-most vertices of a polygon (it is a triangle in the case of this operation gestalt) and the boundary value of a viewing area which a three-dimension body constitutes for every top-most vertices, and carries out the sequential output of the 1-bit flag information according to the comparison result. At this time, sequential storing of the flag information by which a sequential output is carried out whenever one comparison processing is performed is carried out at the shift register 4.

[0038] That is, ALU1 performs the operation shown in the part of (a) in the program shown in drawing 2 . Here, if the run command of compare (a, b) is  $a > b$ , it is the instruction which transmits the flag information on "1" to the shift register



4 of drawing 2 , and transmits "0" when that is not right.

[0039] Subordination connection of three shift registers 4-1 which change by 6 bit patterns corresponding to each triangular top-most vertices, 4-2, and 4-3 is made, and the above-mentioned shift register 4 is constituted. Therefore, whenever it executes one run command of Above compare (a, b), when the above-mentioned instruction is executed 18 times by carrying out the sequential transfer of the flag information on a comparison result at the shift register 4, the top-most-vertices code of each top-most vertices will be stored in three shift registers 4-1, 4-2, and 4-3:

[0040] In addition, according to the logical code (refer to drawing 5 ) currently assigned to each field, it is decided by where [ in the field where each top-most vertices which constitute the above-mentioned triangle like the top-most-vertices code for which it asked in the parts of drawing 4 (a) and (b) were divided into plurality (they are 29 pieces when it is this operation gestalt) ] the top-most-vertices code of each [ these ] top-most vertices belongs.

[0041] 5 is an AND gate means, and it is constituted so that the AND code of "1" or "0" may be outputted by whether there is at least one bit of "1" in each bit which constitutes the AND of the top-most-vertices code of each top-most

vertices stored in the above-mentioned shift register 4. That is, the above-mentioned AND gate means 5 consists of OR gates 5-7 which take the OR of each bit which constitutes the AND searched for by six AND gates 5-1 to 5-6 for taking the AND of the top-most-vertices code of each top-most vertices stored in the above-mentioned shift register 4, and six above-mentioned AND gates 5-1 to 5-6.

[0042] By 1st AND gate 5-1, among six above-mentioned AND gates 5-1 to 5-6 The AND of the least significant bits a0, b0, and c0 of each top-most-vertices code is taken. In the 2nd AND gate (not shown) the 2nd of each top-most-vertices code -- the AND of bit a1, and b1 and c1 -- taking ... in 6th AND gate 5-6, the AND of the most significant bits a5, b5, and c5 of each top-most-vertices code is taken. Thereby, the AND of each top-most-vertices code is outputted by 6 bits from six above-mentioned AND gates 5-1 to 5-6.

[0043] And the OR of each bit which constitutes a 6-bit AND is taken by OR gate 5-7 connected to the latter part of six above-mentioned AND gates 5-1 to 5-6. Therefore, when the AND code of "1" is outputted when there is at least one bit of "1" in each bit which constitutes the AND of each top-most-vertices code, and one does not have the bit of "1", the AND code of "0" comes to be outputted.

Thus, the outputted AND code is stored in AND code register 7b in a flag register 7.

[0044] 6 is an OR-gate means, and it is constituted so that the OR code of "1" or "0" may be outputted by whether there is at least one bit of "1" in each bit which constitutes the OR of the top-most-vertices code of each top-most vertices stored in the above-mentioned shift register 4. That is, the above-mentioned OR-gate means 6 consists of three OR gates 6-1 to 6-3 for taking the OR of each bit which constitutes each top-most-vertices code stored in the above-mentioned shift register 4, and OR gate 6-4 which takes the OR of each bit outputted from three above-mentioned OR gates 6-1 to 6-3.

[0045] In 1st OR gate 6-1 among three above-mentioned OR gates 6-1 to 6-3 The OR of 6 bit a0-a5 which constitute the 1st top-most-vertices code is taken. In the 2nd OR gate (not shown) The OR of 6 bit c0-c5 which constitute the 2nd top-most-vertices code and which constitute the 3rd top-most-vertices code from 3rd OR gate 6-3 is taken by taking 6 bits of ORs of b0-b5. Thereby, the information on a triplet is outputted from three above-mentioned OR gates 6-1 to 6-3.

[0046] And the OR of each bit which constitutes the information on the

above-mentioned triplet is taken by OR gate 6-4 connected to the latter part of three above-mentioned OR gates 6-1 to 6-3. Therefore, when the OR code of "1" is outputted when there is at least one bit of "1" in 18 bits which constitutes the three above-mentioned top-most-vertices codes, and one does not have the bit of "1", the OR code of "0" comes to be outputted. Thus, the outputted OR code is stored in OR code register 7c in a flag register 7.

[0047] In addition, you may make it constitute the above-mentioned OR-gate means 6 from an OR gate which takes the OR of each bit which constitutes the OR searched for by six OR gates for taking the OR of the top-most-vertices code of each top-most vertices stored in the above-mentioned shift register 4 not only like the above configurations but like the AND gate means 5, and the six above-mentioned OR gates.

[0048] in this case -- the 1st OR gate among the six above-mentioned OR gates -- the OR of the least significant bits a0, b0, and c0 of each top-most-vertices code -- taking -- the 2nd OR gate -- the 2nd of each top-most-vertices code -- the OR of bit a1, and b1 and c1 -- taking ... in the 6th OR gate, the OR of the most significant bits a5, b5, and c5 of each top-most-vertices code is taken. Thereby, the OR of each top-most-vertices code is outputted by 6 bits from the six

above-mentioned OR gates.

[0049] And the OR of each bit which constitutes a 6-bit OR is taken by the OR gate connected to the latter part of the six above-mentioned OR gates. Thus, when are constituted, and the OR code of "1" is outputted when there is at least one bit of "1" in 18 bits which constitutes the three above-mentioned top-most-vertices codes, and one does not have the bit of "1", the OR code of "0" comes to be outputted.

[0050] Although it has sign register 7a other than AND code register 7b and OR code register 7c and the flag information showing the results of an operation, such as addition and subtraction in the above ALU 1, having been negative values etc. is stored, this does not characterize the flag register 7 mentioned above in this operation gestalt. It is the AND code and the OR code which are used especially in this operation gestalt. In addition, since such information is acquired with sufficient timing at the time of actuation of the condition judging described below, a physical register is not actually required.

[0051] the invisible field where the corresponding triangle is perfect since I hear that the AND of each top-most-vertices code will not be "000000" if it is the condition judging section, the AND code in the above-mentioned AND code

register 7b (the AND code actually outputted from the above-mentioned AND gate means 5) looks at whether it is "1", and 8 comes out so and there is -- it is (it is out of a viewing area thoroughly) -- it distinguishes.

[0052] the condition judging section 8 -- the visible field where the corresponding triangle is perfect since I hear that the OR of each top-most-vertices code will be "000000" if it sees, it comes out so and it is [ whether the OR code in the above-mentioned OR code register 7c (the OR code actually outputted from the above-mentioned OR-gate means 6) is "0", and ] again -- it is (it is in a viewing area thoroughly) -- it distinguishes.

[0053] moreover, the triangle which corresponds when it judges that the condition judging section 8 fulfills no two above-mentioned conditions -- perfect invisibility or a part -- a visible field -- it is (it is out of a viewing area thoroughly, or the viewing area and the non-display field are straddled) -- it distinguishes.

[0054] Namely, the above-mentioned condition judging section 8 is processing the part of (b) in the program shown in drawing 2 , and it controls it to choose the address value of a branching place by the selector 9 so that it may choose the address value to which only 1 incremented the value of a program counter 10 by the selector 9 according to the distinction result or may perform the next

condition judging.

[0055] As mentioned above, since the shift register 4, the AND gate means 5, and the OR-gate means 6 constituted in hardware the part which searches for the AND of the top-most-vertices code of each top-most vertices, and the part which searches for an OR according to this operation gestalt, when 18 times of comparison operations are performed by ALU1 and the 18th flag information is transmitted to a shift register 4, the AND and OR of each top-most-vertices code come to be searched for automatically. Therefore, the AND and OR of each top-most-vertices code can be searched for promptly, without calculating the part of drawing 4 (c) which ALU31 was performing conventionally.

[0056] Moreover, according to this operation gestalt, the top-most-vertices code of each top-most vertices comes to be called for only by carrying out the sequential transfer of the flag information on a comparison result at the above-mentioned shift register 4, whenever it carries out the sequential comparison of the coordinate value of each top-most vertices, and the boundary value of a viewing area for every top-most vertices as shown in drawing 2 (a), and it performs one comparison processing. Therefore, compared with the conventional operation which also showed the operation of drawing 2 (a) in

ALU1 to drawing 4 (b), it becomes easier, and the initialization processing shown in drawing 4 (a) not only becomes unnecessary, but can ask now for each top-most-vertices code in a easier procedure.

[0057] Furthermore, since according to this operation gestalt OR gate 5-7 and 6-4 constituted in hardware the device for searching for the information referred to as conditions for a conditional-branching instruction when performing field distinction of a graphic form in the condition judging section 8, it comes to ask automatically, without such reference information (the AND code and the OR code) also calculating.

[0058] Namely, AND currently used for a condition judging in the conventional condition judging shown in drawing 4 (d) (and\_code) OR (or\_code) Since it was 6 bits when it was three-dimension space, the 1-bit flag information which adds 0 (10) to the AND and OR, and is acquired as a result needed to be seen. On the other hand, with this operation gestalt, since a condition judging is automatically asked by the hardware-[ the 1-bit usable AND code or the OR code ] configuration, a condition judging can be performed promptly.

[0059] The above result, with this operation gestalt, the number of instructions performed in the case of graphic form field distinction can be reduced to



conventional one half extent, and improvement in the speed of processing can be realized.

[0060]

[Effect of the Invention] As mentioned above, this invention prepares a shift register, transmits the flag information on a comparison result to the above-mentioned shift register for every compare-instruction activation by the comparison means, and asks for each top-most-vertices code. Furthermore, while searching for those ANDs and ORs automatically with the AND gate means and OR-gate means which were constituted in hardware from each top-most-vertices code stored in the above-mentioned shift register since the device which can be made to carry out the direct reference of the result as conditions for a conditional-branching instruction in the case of field distinction was established, in the small amount of hardware, the number of instructions to perform can be markedly boiled compared with the former, and can be lessened, and processing which performs graphic form field distinction can be made into a high speed.

---

## DESCRIPTION OF DRAWINGS

---

[Brief Description of the Drawings]

[Drawing 1] It is drawing showing the example of a configuration of the graphic form field distinction equipment which is 1 operation gestalt of this invention.

[Drawing 2] It is drawing showing the example of the program performed with the graphic form field distinction equipment of this operation gestalt.

[Drawing 3] It is drawing showing the example of a configuration of conventional graphic form field distinction equipment.

[Drawing 4] It is drawing showing the example of the program performed with conventional graphic form field distinction equipment.

[Drawing 5] It is drawing for explaining the graphic form field distinction processing based on a Cohen Sutherland algorithm.

[Description of Notations]

1 ALU

2 Three Register

4 Shift Register

5 AND Gate Means

5-1 to 5-6 AND gate

5-7 OR Gate

6 OR-Gate Means

6-1 to 6-3 OR gate

6-4 OR Gate

7 Flag Register

7b AND code register

7c OR code register

8 Condition Judging Section

9 Selector

10 Program Counter

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-330419

(43)公開日 平成9年(1997)12月22日

(51)Int.Cl. <sup>6</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 T 11/00			G 0 6 F 15/72	3 8 0
G 0 6 F 17/50			G 0 9 G 5/36	5 1 0 V
G 0 9 G 5/36	5 1 0			5 2 0 P
	5 2 0		G 0 6 F 15/60	6 2 6 G

審査請求 未請求 請求項の数6 F D (全 9 頁)

(21)出願番号 特願平8-168421

(22)出願日 平成8年(1996)6月7日

(71)出願人 000006655

新日本製鐵株式会社

東京都千代田区大手町2丁目6番3号

(72)発明者 松浦 一教

東京都千代田区大手町2-6-3 新日本  
製鐵株式会社内

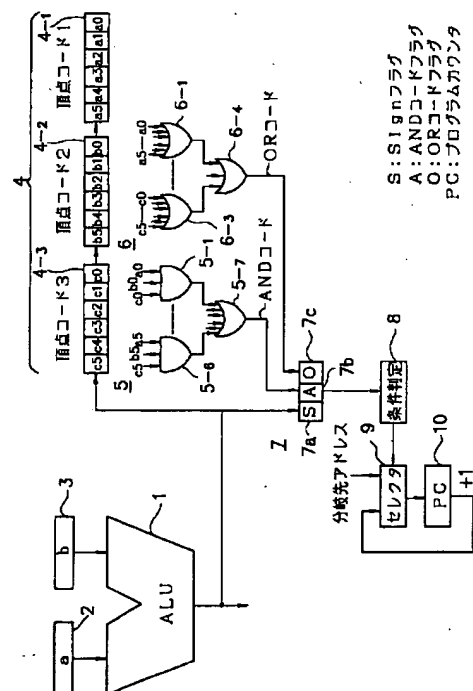
(74)代理人 弁理士 國分 孝悦

(54)【発明の名称】 図形領域判別装置

(57)【要約】

【課題】 Cohen Sutherlandアルゴリズムに基づく図形領域判別処理をより高速に行えるようにする。

【解決手段】 図形を構成する各頂点が属している領域にそれぞれ割り当てられている論理コードを各頂点の頂点コードとして求め、それをシフトレジスタ4に格納するALU1と、上記シフトレジスタ4に格納された各頂点コードの論理積および論理和をとる論理積ゲート手段5および論理和ゲート手段6と、上記論理積ゲート手段5および論理和ゲート手段6の出力情報に基づいて、上記図形がどの領域に存在しているかを判別する条件判定部8とを設け、各頂点コードの論理積を求めると論理和を求めるとをハードウェア的に構成することにより、各頂点コードがシフトレジスタ4に格納された時点で、各頂点コードの論理積と論理和とをALU1で計算を行うことなく直ちに求められるようにする。



## 【特許請求の範囲】

【請求項1】 表示領域とその周囲にある非表示領域とで成る複数の領域中のどこに図形が存在しているかを判別するための図形領域判別装置において、  
上記図形を構成する各頂点が属している領域にそれぞれ割り当てられている論理コードを各頂点の頂点コードとして求め、求めた各頂点の頂点コードをレジスタに格納する頂点コード算出手段と、  
上記頂点コード算出手段により上記レジスタに格納された各頂点の頂点コードの論理積をとる論理積ゲートと、  
上記頂点コード算出手段により上記レジスタに格納された各頂点の頂点コードの論理和をとる論理和ゲートと、  
上記論理積ゲートおよび上記論理和ゲートの出力情報に基づいて、上記図形がどの領域に存在しているかを判別する領域判別手段とを備えたことを特徴とする図形領域判別装置。

【請求項2】 上記頂点コード算出手段は、上記図形を構成する各頂点の座標値と上記複数の領域の中心にある表示領域の境界値とを各頂点ごとに順次比較し、その比較結果に応じたフラグ情報を順次出力する比較手段と、  
上記比較手段より順次出力されるフラグ情報を順次シフトさせながら格納していくことにより上記各頂点の頂点コードを得るシフトレジスタとで構成されることを特徴とする請求項1に記載の図形領域判別装置。

【請求項3】 上記論理積ゲートにより求められた各頂点コードの論理積を構成する各ビットの論理和をとる第1の論理和ゲートを上記論理積ゲートの後段に更に設けるとともに、  
上記論理和ゲートにより求められた各頂点コードの論理和を構成する各ビットの論理和をとる第2の論理和ゲートを上記論理和ゲートの後段に更に設け、  
上記領域判別手段は、上記第1の論理和ゲートの出力情報が“1”であるときには上記図形が完全不可視の領域にあると判別し、上記第2の論理和ゲートの出力情報が“0”であるときには上記図形が完全可視の領域にあると判別し、それ以外のときには上記図形が完全不可視または部分可視の領域にあると判別することを特徴とする請求項1または2に記載の図形領域判別装置。

【請求項4】 表示領域とその周囲にある非表示領域とで成る複数の領域中のどこに図形が存在しているかを判別するための図形領域判別装置において、  
上記図形を構成する各頂点の座標値と上記複数の領域の中心にある表示領域の境界値とを各頂点ごとに順次比較し、その比較結果に応じたフラグ情報を順次出力する比較手段と、  
上記比較手段より順次出力されるフラグ情報を順次シフトさせながら格納していくことにより各頂点の頂点コードを得るシフトレジスタと、  
上記シフトレジスタに格納された各頂点コードの論理積を構成する各ビットの中に“1”のビットが少なくとも

1つあるかどうかによって“1”または“0”のANDコードを出力するように構成した論理積ゲート手段と、  
上記シフトレジスタに格納された各頂点コードの論理和を構成する各ビットの中に“1”のビットが少なくとも1つあるかどうかによって“1”または“0”のORコードを出力するように構成した論理和ゲート手段と、  
上記論理積ゲート手段より出力されるANDコードが“1”であるときには上記図形が完全不可視の領域にあると判別し、上記論理和ゲート手段より出力されるORコードが“0”であるときには上記図形が完全可視の領域にあると判別し、それ以外のときには上記図形が完全不可視または部分可視の領域にあると判別する領域判別手段とを備えたことを特徴とする図形領域判別装置。

【請求項5】 上記図形の存在する空間は、上記表示領域が長方形であり、上記非表示領域が上記表示領域の上下左右方向に対して8個に分割された2次元空間であることを特徴とする請求項1～4の何れか1項に記載の図形領域判別装置。

【請求項6】 上記図形の存在する空間は、上記表示領域が直方体であり、上記非表示領域が上記表示領域の上下左右前後方向に対して28個に分割された3次元空間であることを特徴とする請求項1～4の何れか1項に記載の図形領域判別装置。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 本発明は図形領域判別装置に関し、例えば、コンピュータグラフィックス等の分野で行われるクリッピングの前処理を行う際に用いて好適なものである。

## 【0002】

【従来の技術】 コンピュータグラフィックス(CG)等の分野においては、CGで作られた物体を表示する際にクリッピング処理を行うことが多い。クリッピングとは、表示装置に物体を表示する際に、一定の表示領域を設定しておき、その表示領域の外にはみ出す部分を表示しないようにするために、表示領域の外部にある部分を除外する処理のことを言う。

【0003】 CGの2次元物体や3次元物体を構成するポリゴンが三角形や四角形などの図形によって構成されるが、その図形には、図5に示すように、①完全に表示領域(クリッピングウィンドウ)50の内側にあるもの(完全可視の三角形A)、②完全に表示領域50の外側にあるもの(完全不可視の三角形B、D)、③表示領域50の境界により内側と外側とに分割されるもの(部分可視の三角形C)の3種類がある。この図5の例では、クリッピング処理で、完全不可視の三角形B、D、および部分可視の三角形Cの一部分を除外する処理を行う。

【0004】 このようにクリッピング処理を行うのは、表示されない図形をそのまま残しておく、その後の処理の負荷が大きくなってしまふからである。

【0005】以上のようなクリッピング処理を行うためには、その前処理として、図形が上記①～③のどの領域に存在するのかを判別する必要がある。この3種類の領域を判別するアルゴリズムの1つに、Cohen Sutherlandアルゴリズムと呼ばれるものがある。以下に、2次元CGに適用されるCohen Sutherlandアルゴリズムを図5を用いて説明する。

【0006】まず、クリッピングウィンドウ50を構成するクリッピング直線（4本の境界線）によって分割された9個の領域のそれぞれに、4ビットの論理コードを割り当てる。クリッピングウィンドウ50内の論理コードは“0000”であるが、それよりも左側の領域では最下位ビットに“1”が割り当てられ、右側の領域では第2ビットに“1”が割り当てられる。また、クリッピングウィンドウ50よりも下側の領域では第3ビットに“1”が割り当てられ、上側の領域では最上位ビットに“1”が割り当てられる。

【0007】次に、ポリゴン（図5の例では三角形）の各頂点が上記した9個の領域の中の何れに存在するかを調べ、各頂点に頂点コードを割り当てる。図5の例の場合、三角形A～Dの頂点コードは、それぞれ次のようになる。

三角形A：“0000”、“0000”、“0000”

三角形B：“1001”、“1001”、“1000”

三角形C：“0010”、“0000”、“0010”

三角形D：“1010”、“1000”、“0010”

【0008】そして、上述のようにして割り当てた三角形の各頂点コードを用いて、以下に示す条件判定を行うことにより3つのグループに分類する。

(1) 3つの頂点コードの論理和が“0000”であるなら、その三角形は完全可視である（三角形A）。

(2) 3つの頂点コードの論理積が“0000”でないなら、その三角形は完全不可視である（三角形B）。

(3) 上記(1)(2)以外のものは、完全不可視あるいは部分可視である（三角形Cおよび三角形D）。

【0009】なお、このCohen Sutherlandアルゴリズムでは、上記(2)(3)に示したように、完全不可視の図形と部分可視の図形とを完全には分類することができないが、何れの図形もクリッピング処理の対象となるものであり、正確な分類はその後のクリッピング処理において行われる。以上は2次元空間でのクリッピング前処理であったが、3次元空間では上下左右の要素の他に前後の要素が加わるので、論理コードおよび頂点コードは6ビットになる。

【0010】図3は、上記Cohen Sutherlandアルゴリズムに基づいて図形領域判別を行う従来の図形領域判別装置の構成を示す図であり、図4は、この図形領域判別を行うための従来のプログラムの例を示す図である。以下、これらの図面を用いて説明を行う。

【0011】なお、図4のプログラムは、通常のプロセ

ッサ（DSPなど）で行うプログラムを示したものであり、クリッピングウィンドウ50は、

$$-W \leq X \leq W, -W \leq Y \leq W, -W \leq Z \leq W$$

の範囲でなる立方体であると仮定している。ただし、X、Y、Zはそれぞれクリッピングウィンドウ50の中心を始点とした左右方向、上下方向、前後方向の座標値である。また、処理中のポリゴンの各頂点の座標を（X[i], Y[i], Z[i]）としている（i=0, 1, 2）。

【0012】図4において、(a)の部分は、三角形を構成する3頂点の頂点コードvertex[i]（i=0,1,2）を全て0<sub>(10)</sub> = 000000<sub>(2)</sub> に初期化する処理である。また、(b)の部分は、各頂点コードvertex[i]を実際に求める処理である。3次元空間で図形領域判別を行う場合、1つの頂点コードを求めるのに6回の条件判定を行う必要がある。

【0013】1回目の条件判定は、ある頂点のX座標値X[i]がクリッピングウィンドウ50の左側の境界値である-Wよりも小さいか否かを判定する処理であり、この条件が成立する場合は、そのときの頂点コードvertex[i]（初期化されたもの）と1<sub>(10)</sub> = 000001<sub>(2)</sub>との論理和をとったものが新たな頂点コードvertex[i]とされることにより、最下位ビットに“1”が立てられる。

【0014】2回目の条件判定は、上記X座標値X[i]がクリッピングウィンドウ50の右側の境界値であるWよりも大きいのか否かを判定する処理であり、この条件が成立する場合は、そのときの頂点コードvertex[i]と2<sub>(10)</sub> = 000010<sub>(2)</sub>との論理和をとったものが新たな頂点コードvertex[i]とされることにより、2ビット目に“1”が立てられる。

【0015】以下同様にして、Y座標値Y[i]、Z座標値Z[i]についての条件判定を順次行う。これら6回に渡る条件判定では、記号orで示すように直前の頂点コードvertex[i]の値とで論理和をとっているため、成立する条件式に対応するビットの全てに“1”が立てられる。この6回の条件判定処理により、1つの頂点についての頂点コードvertex[i]が求められる。3次元物体を構成するポリゴンが三角形の場合、このような処理を3回繰り返して行うことにより、3つの頂点コードvertex[i]（i=0,1,2）が求められる。

【0016】このようにして頂点コードvertex[i]を求める図4(a)(b)の処理は、図3のALU（算術論理演算装置）31が行っている。このとき、6回の条件判定処理で用いている境界値W、-Wは、例えば一方のレジスタ32に格納され、各座標値X[i]、Y[i]、Z[i]は他方のレジスタ33に格納される。

【0017】また、図4(c)(d)の部分は、上述のようにして求めた三角形の各頂点コードvertex[i]（i=0,1,2）を用いて、上記した(1)～(3)のグループ分けを行う処理である。すなわち、図4(c)の部分では、

3つの頂点コードvertex[0], vertex[1], vertex[2]の論理積と論理和とを求め、図4(d)の部分では、その求めた論理積が0<sub>(10)</sub>でないかどうか、および求めた論理和が0<sub>(10)</sub>であるかどうかを見ることによって3種類のグループ分けを行う。

【0018】上記図4(c)の処理は図3のALU31が行い、図4(d)の処理は図3のALU31および条件判定部35が行っている。すなわちALU31は、まず、図4(b)の処理によって求めた3つの頂点コードvertex[i] (i=0,1,2)をレジスタ32, 33を介して入力し、それらの論理積と論理和とを求める。次に、求めた論理積の値が0<sub>(10)</sub>でないか、すなわち、該当するポリゴン(三角形)が上記(2)のグループに属するものであるかどうかを調べる。

【0019】なお、図4(b)の計算の性質上、論理積が負の値をとることはないから、論理積が0<sub>(10)</sub>でないことは、論理積が0<sub>(10)</sub>よりも大きいということと同じである。したがって、図4(d)の最上段に示したように、求めた論理積が0<sub>(10)</sub>より大きいかどうかを見ることによって上記(2)のグループに属するかどうかを以下のようにして判定している。

【0020】まず、ALU31が、上記求めた論理積の値に対して0<sub>(10)</sub>を加算する。このとき、加算処理により得られた値が0<sub>(10)</sub>であればサインフラグレジスタ34にゼロフラグの“1”が立てられる。よって、サインフラグレジスタ34の値が“0”であれば、求めた論理積は0<sub>(10)</sub>より大きいということになる。そこで、条件判定部35は、上記サインフラグレジスタ34の値を見ることによって、求めた論理積が0<sub>(10)</sub>より大きいかどうかを調べる。

【0021】条件判定部35は、上記サインフラグレジスタ34の値が“0”であると判断したときは、求めた論理積の値が0<sub>(10)</sub>ではないということなので、プログラムカウンタ37の値を1だけインクリメントしたアドレス値をセレクト36で選択するように制御し、該当する三角形が完全不可視のものであるとする処理を行う。一方、上記サインフラグレジスタ34の値が“1”であると判断したときは、次に論理和に関する条件判定を行うべく分岐先のアドレス値をセレクト36で選択するように制御する。

【0022】このようにして論理和の条件判定を行う分岐先アドレスにプログラムが進められると、次に、ALU31および条件判定部35は、図4(d)の第2段目に示したように、求めた論理和の値が0<sub>(10)</sub>であるかどうか、すなわち、該当するポリゴン(三角形)が上記(1)のグループに属するものであるかどうかを調べる。この場合も、ALU31において論理和の値に対して0<sub>(10)</sub>を加算したときに得られるサインフラグレジスタ34の値を条件判定部35が見ることによって、求めた論理和の値が0<sub>(10)</sub>であるかどうかを調べる。

【0023】条件判定部35は、上記サインフラグレジスタ34の値が“1”であると判断したときは、求めた論理和の値が0<sub>(10)</sub>であるということなので、プログラムカウンタ37の値を1だけインクリメントしたアドレス値をセレクト36で選択するように制御し、該当する三角形が完全可視のものであるとする処理を行う。

【0024】一方、上記サインフラグレジスタ34の値が“0”であると判断したときは、図4(d)の最下段の処理を行うべく分岐先のアドレス値をセレクト36で選択するように制御し、その分岐先にプログラムを進めて、該当する三角形が完全不可視または部分可視のものであるとする処理を行う。

【0025】

【発明が解決しようとする課題】図4に示したように、従来の図形領域判別装置では、Cohen Sutherlandアルゴリズムに基づいてポリゴンが何れの領域に存在しているのかを判別するために、数多くの命令を実行している。そのため、上記アルゴリズムの実行に長い時間がかかっていた。そこで本発明は、上記Cohen Sutherlandアルゴリズムに基づく図形領域判別処理をより高速に行えるようにすることを目的とする。

【0026】

【課題を解決するための手段】本発明の図形領域判別装置は、表示領域とその周囲にある非表示領域とで成る複数の領域中のどこに図形が存在しているかを判別するための図形領域判別装置において、上記図形を構成する各頂点が属している領域にそれぞれ割り当てられている論理コードを各頂点の頂点コードとして求め、求めた各頂点の頂点コードをレジスタに格納する頂点コード算出手段と、上記頂点コード算出手段により上記レジスタに格納された各頂点の頂点コードの論理積をとる論理積ゲートと、上記頂点コード算出手段により上記レジスタに格納された各頂点の頂点コードの論理和をとる論理和ゲートと、上記論理積ゲートおよび上記論理和ゲートの出力情報に基づいて、上記図形がどの領域に存在しているかを判別する領域判別手段とを備えたことを特徴とする。

【0027】本発明の他の特徴とするところは、上記頂点コード算出手段は、上記図形を構成する各頂点の座標値と上記複数の領域の中心にある表示領域の境界値とを各頂点ごとに順次比較し、その比較結果に応じたフラグ情報を順次出力する比較手段と、上記比較手段より順次出力されるフラグ情報を順次シフトさせながら格納していくことにより上記各頂点の頂点コードを得るシフトレジスタとで構成されることを特徴とする。

【0028】本発明のその他の特徴とするところは、上記論理積ゲートにより求められた各頂点コードの論理積を構成する各ビットの論理和をとる第1の論理和ゲートを上記論理積ゲートの後段に更に設けるとともに、上記論理和ゲートにより求められた各頂点コードの論理和を構成する各ビットの論理和をとる第2の論理和ゲートを



上記論理和ゲートの後段に更に設け、上記領域判別手段は、上記第1の論理和ゲートの出力情報が“1”であるときには上記図形が完全不可視の領域にあると判別し、上記第2の論理和ゲートの出力情報が“0”であるときには上記図形が完全可視の領域にあると判別し、それ以外のときには上記図形が完全不可視または部分可視の領域にあると判別することを特徴とする。

【0029】本発明のその他の特徴とするところは、表示領域とその周囲にある非表示領域とで成る複数の領域中のどこに図形が存在しているかを判別するための図形領域判別装置において、上記図形を構成する各頂点の座標値と上記複数の領域の中心にある表示領域の境界値とを各頂点ごとに順次比較し、その比較結果に応じたフラグ情報を順次出力する比較手段と、上記比較手段より順次出力されるフラグ情報を順次シフトさせながら格納していくことにより各頂点の頂点コードを得るシフトレジスタと、上記シフトレジスタに格納された各頂点コードの論理積を構成する各ビットの中に“1”のビットが少なくとも1つあるかどうかによって“1”または“0”のANDコードを出力するように構成した論理積ゲート手段と、上記シフトレジスタに格納された各頂点コードの論理和を構成する各ビットの中に“1”のビットが少なくとも1つあるかどうかによって“1”または“0”のORコードを出力するように構成した論理和ゲート手段と、上記論理積ゲート手段より出力されるANDコードが“1”であるときには上記図形が完全不可視の領域にあると判別し、上記論理和ゲート手段より出力されるORコードが“0”であるときには上記図形が完全可視の領域にあると判別し、それ以外のときには上記図形が完全不可視または部分可視の領域にあると判別する領域判別手段とを備えたことを特徴とする。

【0030】本発明のその他の特徴とするところは、上記図形の存在する空間は、上記表示領域が長方形であり、上記非表示領域が上記表示領域の上下左右方向に対して8個に分割された2次元空間であることを特徴とする。

【0031】本発明のその他の特徴とするところは、上記図形の存在する空間は、上記表示領域が直方体であり、上記非表示領域が上記表示領域の上下左右前後方向に対して28個に分割された3次元空間であることを特徴とする。

【0032】本発明は上記技術手段より成るので、各頂点の頂点コードの論理積を求める部分と論理和を求める部分とがそれぞれ論理積ゲートおよび論理和ゲートによりハードウェア的に構成され、各頂点の頂点コードが求められた時点でそれらの論理積と論理和とが自動的に直ちに求められるようになり、論理積や論理和を従来のように計算によって求めなくても済む。

【0033】本発明の他の特徴によれば、各頂点の頂点コードは、各頂点の座標値と表示領域の境界値とを各頂

点ごとに順次比較し、比較処理を1つ実行するごとに比較結果のフラグ情報をシフトレジスタに順次転送するだけで求められるので、従来に比べてより少ない計算手順によって各頂点コードが求められるようになる。

【0034】本発明のその他の特徴によれば、領域判別を行う際に条件分岐命令の条件として参照する情報を求めるための機構が第1、第2の論理和ゲートによりハードウェア的に構成されているので、そのような参照情報が各頂点コードの論理積および論理和から自動的に直ちに求められるようになり、従来のように計算によって求めなくても済む。

【0035】

【発明の実施の形態】以下、本発明の一実施形態を図面に基づいて説明する。図1は、表示領域とその周囲にある複数の分割された非表示領域とで成る複数の領域中のどこに図形が存在しているかを判別するための本実施形態に係る図形領域判別装置の構成を示す図であり、図2は、この図形領域判別を行うための本実施形態によるプログラムの例を示す図である。

【0036】なお、図2のプログラムでは、表示領域（クリッピングウィンドウ）が、 $-W \leq X \leq W$ ,  $-W \leq Y \leq W$ ,  $-W \leq Z \leq W$ の範囲でなる立方体であると仮定している。ただし、 $X$ ,  $Y$ ,  $Z$ はそれぞれクリッピングウィンドウの中心を始点とした左右方向、上下方向、前後方向の座標値である。また、処理中のポリゴンの各頂点の座標を( $X[i]$ ,  $Y[i]$ ,  $Z[i]$ )としている( $i=0, 1, 2$ )。この場合、非表示領域は、表示領域の上下左右前後方向に28個の領域に分割される。

【0037】図1において、1はALUであり、2つのレジスタ2, 3に格納される情報を用いて以下に述べるような演算を行う。すなわち、ALU1は、3次元物体の構成するポリゴン（本実施形態の場合は三角形）の各頂点の座標値と表示領域の境界値とを各頂点ごとに順次比較し、その比較結果に応じた1ビットのフラグ情報を順次出力する演算を行う。このとき、1つの比較処理が行われるごとに順次出力されるフラグ情報は、シフトレジスタ4に順次格納されていく。

【0038】つまり、ALU1は、図2に示したプログラム中の(a)の部分に示した演算を行う。ここで、compare(a,b)の実行命令は、 $a > b$ であれば、図2のシフトレジスタ4に“1”のフラグ情報を転送し、そうでない場合は“0”を転送するようにする命令である。

【0039】上記シフトレジスタ4は、三角形の各頂点に対応した6ビット構成で成る3つのシフトレジスタ4-1, 4-2, 4-3が従属接続されて構成されている。したがって、上記compare(a,b)の実行命令を1つ実行するごとに比較結果のフラグ情報をシフトレジスタ4に順次転送していくことにより、上記命令を18回実行した時点で3つのシフトレジスタ4-1, 4-2, 4-3に各頂点の頂

点コードが格納されることとなる。

【0040】なお、これら各頂点の頂点コードは、図4 (a) (b) の部分で求めた頂点コードと同様に、上記三角形を構成する各頂点が複数（本実施形態の場合は29個）に分割された領域中のどこに属しているかによって、それぞれの領域に割り当てられている論理コード（図5参照）に応じて決められるものである。

【0041】5は論理積ゲート手段であり、上記シフトレジスタ4に格納された各頂点の頂点コードの論理積を構成する各ビットの中に“1”のビットが少なくとも1つあるかどうかによって“1”または“0”のANDコードを出力するように構成されている。すなわち、上記論理積ゲート手段5は、上記シフトレジスタ4に格納された各頂点の頂点コードの論理積をとるための6個の論理積ゲート5<sub>1</sub>～5<sub>6</sub>と、上記6個の論理積ゲート5<sub>1</sub>～5<sub>6</sub>により求められた論理積を構成する各ビットの論理和をとる論理和ゲート5<sub>7</sub>とで構成される。

【0042】上記6個の論理積ゲート5<sub>1</sub>～5<sub>6</sub>のうち、1番目の論理積ゲート5<sub>1</sub>では、各頂点コードの最下位ビットa0、b0、c0の論理積をとり、2番目の論理積ゲート（図示せず）では、各頂点コードの第2ビットa1、b1、c1の論理積をとり、・・・6番目の論理積ゲート5<sub>6</sub>では、各頂点コードの最上位ビットa5、b5、c5の論理積をとる。これにより、上記6個の論理積ゲート5<sub>1</sub>～5<sub>6</sub>から各頂点コードの論理積が6ビットで出力される。

【0043】そして、上記6個の論理積ゲート5<sub>1</sub>～5<sub>6</sub>の後段に接続された論理和ゲート5<sub>7</sub>により、6ビットの論理積を構成する各ビットの論理和がとられる。したがって、各頂点コードの論理積を構成する各ビットの中に、“1”のビットが少なくとも1つあるときには“1”のANDコードが出力され、また“1”のビットが1つもないときには“0”のANDコードが出力されるようになる。このようにして出力されたANDコードは、フラグレジスタ7中のANDコードレジスタ7bに格納される。

【0044】6は論理和ゲート手段であり、上記シフトレジスタ4に格納された各頂点の頂点コードの論理和を構成する各ビットの中に“1”のビットが少なくとも1つあるかどうかによって“1”または“0”のORコードを出力するように構成されている。すなわち、上記論理和ゲート手段6は、上記シフトレジスタ4に格納された各頂点コードを構成する各ビットの論理和をとるための3個の論理和ゲート6<sub>1</sub>～6<sub>3</sub>と、上記3個の論理和ゲート6<sub>1</sub>～6<sub>3</sub>から出力される各ビットの論理和をとる論理和ゲート6<sub>4</sub>とで構成される。

【0045】上記3個の論理和ゲート6<sub>1</sub>～6<sub>3</sub>のうち、1番目の論理和ゲート6<sub>1</sub>では、第1の頂点コードを構成する6ビットa0～a5の論理和をとり、2番目の論理和ゲート（図示せず）では、第2の頂点コードを

構成する6ビットb0～b5の論理和をとり、3番目の論理和ゲート6<sub>3</sub>では、第3の頂点コードを構成する6ビットc0～c5の論理和をとる。これにより、上記3個の論理和ゲート6<sub>1</sub>～6<sub>3</sub>から3ビットの情報が出力される。

【0046】そして、上記3個の論理和ゲート6<sub>1</sub>～6<sub>3</sub>の後段に接続された論理和ゲート6<sub>4</sub>により、上記3ビットの情報を構成する各ビットの論理和がとられる。したがって、上記3個の頂点コードを構成する18ビットの中に、“1”のビットが少なくとも1つあるときには“1”のORコードが出力され、また“1”のビットが1つもないときには“0”のORコードが出力されるようになる。このようにして出力されたORコードは、フラグレジスタ7中のORコードレジスタ7cに格納される。

【0047】なお、上記論理和ゲート手段6は、上記のような構成に限らず、論理積ゲート手段5と同様に、上記シフトレジスタ4に格納された各頂点の頂点コードの論理和をとるための6個の論理和ゲートと、上記6個の論理和ゲートにより求められた論理和を構成する各ビットの論理和をとる論理和ゲートとで構成するようにしてもよい。

【0048】この場合には、上記6個の論理和ゲートのうち、1番目の論理和ゲートでは、各頂点コードの最下位ビットa0、b0、c0の論理和をとり、2番目の論理和ゲートでは、各頂点コードの第2ビットa1、b1、c1の論理和をとり、・・・6番目の論理和ゲートでは、各頂点コードの最上位ビットa5、b5、c5の論理和をとる。これにより、上記6個の論理和ゲートから各頂点コードの論理和が6ビットで出力される。

【0049】そして、上記6個の論理和ゲートの後段に接続された論理和ゲートにより、6ビットの論理和を構成する各ビットの論理和がとられる。このように構成した場合にも、上記3個の頂点コードを構成する18ビットの中に“1”のビットが少なくとも1つあるときには“1”のORコードが出力され、また“1”のビットが1つもないときには“0”のORコードが出力されるようになる。

【0050】上述したフラグレジスタ7は、ANDコードレジスタ7bおよびORコードレジスタ7cの他に、サインレジスタ7aを備えており、上記ALU1における加減算などの演算結果が負の値であったことを表すフラグ情報などが格納されるが、これは本実施形態において特徴的なものではない。本実施形態において特に利用するのはANDコードおよびORコードである。なお、これらの情報は以下に述べる条件判定の動作時にタイミングよく得られるので、実際には物理的なレジスタは必要ではない。

【0051】8は条件判定部であり、上記ANDコードレジスタ7b内のANDコード（実際には上記論理積ゲ

10

20

30

40

50

ート手段5より出力されるANDコード)が“1”であるかどうかを見て、そうであれば、各頂点コードの論理積が“000000”ではないということなので、該当する三角形は完全不可視の領域にある(完全に表示領域外にある)と判別する。

【0052】条件判定部8また、上記ORコードレジスタ7c内のORコード(実際には上記論理和ゲート手段6より出力されるORコード)が“0”であるかどうかを見て、そうであれば、各頂点コードの論理和が“000000”であるということなので、該当する三角形は完全可視の領域にある(完全に表示領域内にある)と判別する。

【0053】また、条件判定部8は、上記2つの条件を何れも満たさないと判断したときには、該当する三角形は完全不可視または部分可視の領域にある(完全に表示領域外にあるか表示領域と非表示領域とにまたがっている)と判別する。

【0054】すなわち、上記条件判定部8は、図2に示したプログラム中の(b)の部分の処理を行っており、その判別結果に応じて、プログラムカウンタ10の値を1だけインクリメントしたアドレス値をセクタ9で選択するか、あるいは次の条件判定を行うべく分岐先のアドレス値をセクタ9で選択するように制御する。

【0055】以上のように、本実施形態によれば、各頂点の頂点コードの論理積を求める部分と論理和を求める部分とをシフトレジスタ4、論理積ゲート手段5および論理和ゲート手段6によりハードウェア的に構成したので、ALU1により18回の比較演算が実行されて18個目のフラグ情報がシフトレジスタ4に転送された時点で、各頂点コードの論理積と論理和とが自動的に求められるようになる。したがって、従来ALU31が行っていた図4(c)の部分の計算を行うことなく、各頂点コードの論理積や論理和を直ちに求めることができる。

【0056】また、本実施形態によれば、各頂点の頂点コードは、図2(a)に示したように各頂点の座標値と表示領域の境界値とを各頂点ごとに順次比較して、比較処理を1つ実行するごとに比較結果のフラグ情報を上記シフトレジスタ4に順次転送するだけで求められるようになる。したがって、図4(a)に示した初期化処理が不要になるだけでなく、ALU1における図2(a)の演算も図4(b)に示した従来の演算に比べてより簡単になり、各頂点コードをより簡単な手続きで求めることができるようになる。

【0057】さらに、本実施形態によれば、条件判定部8で図形の領域判別を行う際に条件分岐命令の条件として参照する情報を求めるための機構を論理和ゲート5<sub>1</sub>、6<sub>1</sub>によりハードウェア的に構成したので、そのような参照情報(ANDコードおよびORコード)も計算を行うことなく自動的に求められるようになる。

【0058】すなわち、図4(d)に示した従来の条件

判定では、条件判定に使用している論理積(and\_code)や論理和(or\_code)が3次元空間の場合には6ビットであるので、その論理積や論理和に0<sub>(10)</sub>を加算してその結果得られる1ビットのフラグ情報を見る必要があった。これに対して本実施形態では、条件判定に使用可能な1ビットのANDコードやORコードがハードウェア的な構成により自動的に求められるので、条件判定を直ちに行うことができるようになる。

【0059】以上の結果、本実施形態では、図形領域判別の際に実行する命令数を従来の半分程度に削減することができ、処理の高速化を実現することができる。

#### 【0060】

【発明の効果】本発明は上述したように、シフトレジスタを設け、比較手段による比較命令実行ごとに比較結果のフラグ情報を上記シフトレジスタに転送して各頂点コードを求める。さらに、上記シフトレジスタに格納した各頂点コードからそれらの論理積と論理和とをハードウェア的に構成された論理積ゲート手段および論理和ゲート手段により自動的に求めるとともに、その結果を領域判別の際に条件分岐命令の条件として直接参照できるようにする機構を設けたので、少ないハードウェア量で、実行する命令数を従来に比べて格段に少なくすることができ、図形領域判別を行う処理を高速にすることができる。

#### 【図面の簡単な説明】

【図1】本発明の一実施形態である図形領域判別装置の構成例を示す図である。

【図2】本実施形態の図形領域判別装置で実行するプログラムの例を示す図である。

【図3】従来の図形領域判別装置の構成例を示す図である。

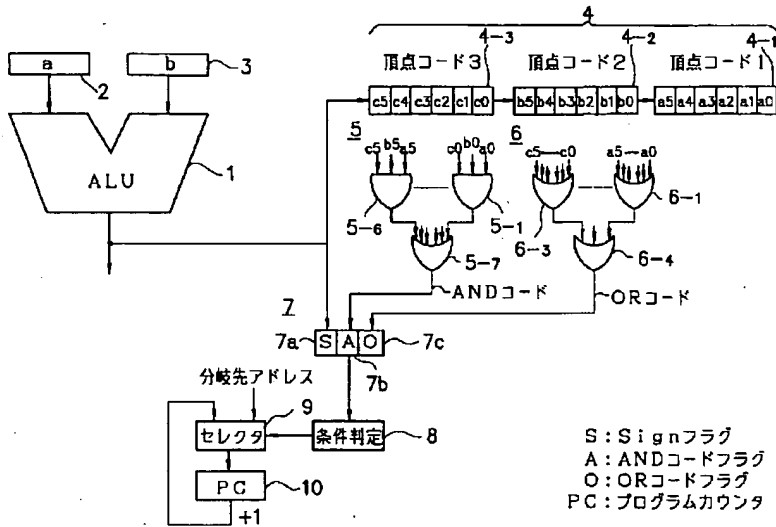
【図4】従来の図形領域判別装置で実行するプログラムの例を示す図である。

【図5】Cohen Sutherlandアルゴリズムに基づく図形領域判別処理を説明するための図である。

#### 【符号の説明】

- 1 ALU
- 2, 3 レジスタ
- 4 シフトレジスタ
- 5 論理積ゲート手段
- 5<sub>1</sub> ~ 5<sub>4</sub> 論理積ゲート
- 5<sub>7</sub> 論理和ゲート
- 6 論理和ゲート手段
- 6<sub>1</sub> ~ 6<sub>3</sub> 論理和ゲート
- 6<sub>4</sub> 論理和ゲート
- 7 フラグレジスタ
- 7b ANDコードレジスタ
- 7c ORコードレジスタ
- 8 条件判定部
- 9 セレクタ

【図1】



【図2】

```

for i=0 to 2 begin
  compare (-W, X[i]);
  compare (X[i], W);
  compare (-W, Y[i]);
  compare (Y[i], W);
  compare (-W, Z[i]);
  compare (Z[i], W);
end;

```

(a)

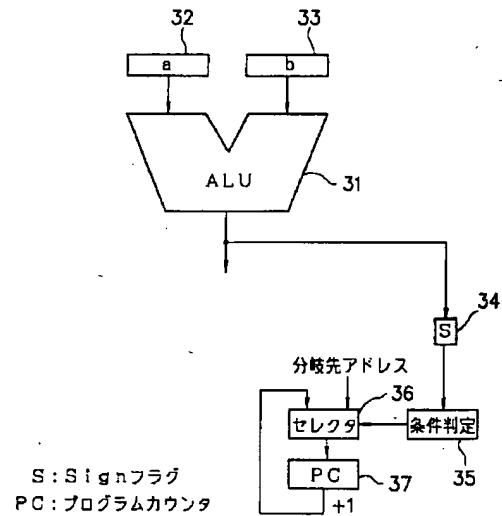
```

if ANDコード=1 then ポリゴンは完全不可視
else if ORコード=0 then ポリゴンは完全可視
else ポリゴンは完全不可視か部分可視

```

(b)

【図3】



【図4】

```

vertex[0]=0;
vertex[1]=0;
vertex[2]=0;
} (a)

for i = 0 to 2 begin
  if X[i] < -W then vertex[i]=vertex[i] or 1;
  if X[i] > W then vertex[i]=vertex[i] or 2;
  if Y[i] < -W then vertex[i]=vertex[i] or 4;
  if Y[i] > W then vertex[i]=vertex[i] or 8;
  if Z[i] < -W then vertex[i]=vertex[i] or 16;
  if Z[i] > W then vertex[i]=vertex[i] or 32;
} (b)
end;

and_code=vertex[0] and vertex[1] and vertex[2];
or_code=vertex[0] or vertex[1] or vertex[2];
} (c)

if and_code > 0 then ポリゴンは完全不可視
else if or_code = 0 then ポリゴンは完全可視
else ポリゴンは完全不可視, または部分可視;
} (d)

```

【図5】

